# Cost and Energy Reduction Evaluation for ARM Based Web Servers

Olle Svanfeldt-Winter, Sébastien Lafond, Johan Lilius
*Åbo Akademi University*
*Department of Information Technologies*
*Joukahainengatan 3-5, 20520 Turku, Finland*
*Email: osvanfel@abo.fi*

***Abstract*—Direct energy and energy dependent infrastructure costs are major contributors to the total cost of a data center. This paper evaluates the energy saving potential of the ARM Cortex-A9 MPCore processor, compared to more conventional Intel Xeon processors, when running certain server applications. The presented measurements show a three to eleven times better energy efficiency for ARM Cortex-A9 processors compared to Intel Xeon processors. Using these measurements as input to the cost model for a typical datacenter presented by Hamilton in [1], [2], we analyze the resulting cost reductions.**

***Keywords*-Cloud Computing, Energy Efficiency, Apache HTTP server, ARM, SIP-Proxy, Erlang**

## I. Introduction

Cloud computing systems often use large server farms in order to provide services. These server farms have high energy consumption and energy is not only needed to operate the servers themselves, but also to operate the required cooling infrastructure. Energy consumption is seen as both an economical and ecological issue. The approach presented in this paper aims at evaluating the obtainable cost and energy dissipation reduction, if energy efficient CPUs, like the ARMv7 based ARM Cortex-A9 MPCore processor, would be used in a server farm. The architecture of processors used in smartphones and embedded systems have been designed with energy efficiency in mind from the beginning, something that has not been the case with the x86 architecture, usually found in servers. This makes these processors interesting candidates when looking for replacements, to regular x86 architecture based server processors.

The computational power of a single low power processor is generally modest compared to traditional desktop and server processors. The price difference between server grade x86 processors typically used in servers, and the relatively small Cortex-A9 processors, will not be taken into account as no price information is available. Moving to processors with lower individual performance increases the number of processors needed to provide the same service as before. Distributing work on a larger number of processors increases the importance of parallelism on the software side. Applications designed to be run on server farms are, however, already designed to be distributable in order to use the added resources from a server farm.

We will begin by looking at the contribution from the CPU to the total server energy consumption, and the performance unbalance between computer components. The total server energy consumption for a complete data center will then be analyzed. We will continue by presenting the benchmarks and the test hardware that will be used in the measurements, followed by evaluations of the achieved results, and finally ending with conclusions.

In [1], [2] Hamilton introduced a model for a hypothetical data center and establish the relationship between the total cost of a data center and the energy consumption of its servers. In this paper we compare the long term investment costs, the infrastructure and the servers costs, to the cost of energy. Hamilton concludes that the cost of energy is 13 % and the cost for energy dependent infrastructure is 18 % of the total cost of a typical data center. Together these energy related costs amounts to 31 % of the total data center cost.

In this paper we evaluate the energy efficiency of two ARM Cortex-A9 MPCore processors, compared to Intel Xeon processors, for typical server applications. The obtained results are then used as input to the cost model in [2] to estimate the achievable energy cost savings for a data center using ARM Cortex-A9 MPCore processors.

## II. Energy proportional computing

In an energy proportional server the energy consumption would be in direct relation to the required performance. As Barroso and Holzle states in [3] this is far from a reality in moderns servers. The energy efficiency of a server is best at peak performance, although a typical server operates most of the time between 10 and 50 % of their maximum capacity. Even when idling a server uses close to half of its peak power consumption, in a completely energy proportional server, no energy would be used while idling. In practice total energy proportionality cannot be achieved due to leakage currents in todays CMOS technology. Processors designed for embedded systems has clearly better energy proportionality than processors designed for servers.

David A. Patterson points out in [4] a problem with performance unbalance between CPUs and the rest of the components in a computer. The performance improvement of CPUs has been faster than for other components. Bandwidth between components such as the CPU and memory can be improved by adding more communications paths between them, but it is costly and causes an increase in energy consumption and the size of the circuits. The performance difference between the CPU and the rest of the components hinders effective usage of the available CPU resources. As Hamilton [1] points out there are at least two ways of dealing with the performance unbalance problem. One is to simply invest in better bandwidth and communication paths between the memory and CPU. Another way is to avoid the problem by using cheaper, lower-powered CPUs that do not need fast memory. Hamilton also points out that as server hardware is built with higher quality requirements, and in lower volumes than client hardware, they are more expensive. Hamilton continues that ”When we replace servers well before they fail, we are effectively paying for quality that we’re not using“. The energy efficiency is in general better for newer hardware, adding to the pressure to upgrade to newer servers. In this paper we explore the potential cost benefits achievable by the usage of an increased number of cheaper, more efficient processors.

## III. Energy consumption

The biggest consumer of energy in a server is the CPU. Barroso and Holzle [3] states that the processor in a server used by Google in 2007, contributed with approximately 45 % of the server total power at peak performance, and approximately 27 % when idle. The energy consumption ratios between different components vary depending on the configuration of the server. Fan et al. [5] shows that in servers with several disk drives the energy consumption of the disk drives also becomes significant. The power consumption for a server is application specific and show that in a typical data center the consumption is 72 % of the actual peak power consumption. Regardless of the average power dissipation during standard operation a data center must still have the infrastructure to support the maximum power that the servers can dissipate. Reducing the peak power dissipation also reduces the demand on the power and cooling infrastructure. Fan et al. conclude that peak power consumption is the most important factor for guiding server deployment in data centers, but that the energy bill is defined by the average consumption. With a lower peak power consumption a larger number of servers can be used within the same energy budget, leading to a higher utilization level of the cooling and power infrastructure, and thereby a more effective use of the available resources and budget. The requirements for both cooling and power, including UPS are reduced with lower peak power dissipation.

The two main metrics for data center energy efficiency, Power Usage Effectiveness (PUE) and Data Center Infrastructure Efficiency (DCiE) are defined by the Green Grid [6] as:

$$PUE = \frac{TotalFacilityPower}{ITEquipmentPower}$$

$$DCiE = \frac{1}{PUE} = \frac{ITEquipmentPower}{TotalFacilityPower} \times 100$$

IT Equipment power includes the servers but also network equipments and equipment used to monitor and control the data center. Total facility power includes in addition to the IT equipment the cooling, UPS, lighting and distribution losses external to the IT equipment. In an ideal data center the PUE would be 1, and would mean that all power used by the center is used to power the IT equipment. According to [6], preliminary data shows that many data centers have a PUE of 3.0 or greater.

Several companies including Google and Microsoft are building data centers with increasing numbers of servers [7], [8]. Many of the centers are so large that instead of using a server rack as the basic unit, shipping containers are used. Google’s server containers are reported to house 1160 servers, and the power dissipation of just one container is reported to be up to 250 KW. Which means approximately 216 W per server.

In 2008 Microsoft announced that they were building a data center containing 300 000 servers [9]. If the power dissipation of the servers in Microsoft’s new server farm is the same as the one reported by Google, the power consumption of the servers in the farm would approximately be 65 MW. The fact that the servers are packed tightly also means that the challenges for the cooling is increasing. The problem with large heat dissipation is being addressed in different ways, for example Intel provides energy efficient versions of some of its Xeon processors, intended especially for high density blade servers [10]. The more energy efficient versions are also generally more expensive. Having to pay less for keeping the servers running and still providing the same services makes new business opportunities possible, and increases the profit in current business areas.

## IV. Data center cost

In order to evaluate the potential savings enabled by a reduction of the energy consumption, the total cost structure for a server farm must be analyzed. Hamilton [1], [2] presents a cost analysis for a hypothetical data center to enable the comparison between cost elements such as infrastructure, servers and power. Amortization times are also defined for the investments. The infrastructure in Hamilton’s hypothetical data center is designed to have a 10-year amortization time, a 4-year amortization time for networking equipment, and a 3-year amortization time for the servers. A five percent yearly cost for the capital used to fund the data center is added, and the cost of energy is set at $0.07 per KWh. An 80 % average critical load usage is assumed and a server is assumed to dissipate 165 W. The cost structure of the data center can be seen in the pie chart shown in Figure 1. The chart shows that the direct cost contribution of energy to the total data center cost is 13 %. Hamilton continues to point out that while this is not a huge percentage, the energy consumption also has indirect impact on cost as the maximum power consumption of the servers is reflected in the infrastructure costs. For the hypothetical data center 18 % of total data center cost consists of power and cooling infrastructure. Improved energy efficiency will have a strong impact on this cost. In the hypothetical data center the cost of power and cooling infrastructure, combined with the actual power usage is then 31 %.

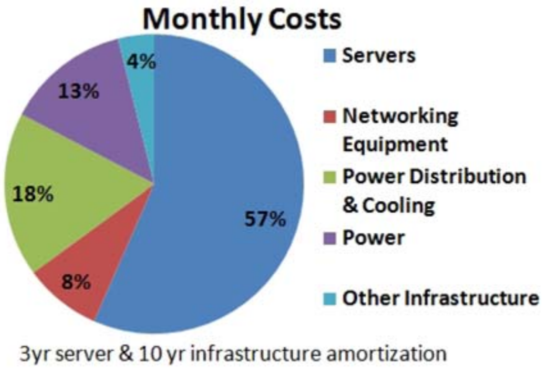

Figure 1. Monthly costs for server, power and infrastructure [2]

## V. Evaluated hardware

### A. Versatile Express

The evaluation of the four core ARM Cortex-A9 processor was done using the Versatile Express [11] development platform. The Versatile Express consists of a motherboard (V2M-P1) that supports the simultaneous evaluation of two daughterboards. CoreTile Express A9 MPCore (V2P-CA9) is the daughter board that will be used for this evaluation. The daughter board has 1 GB of DDR2 memory, and an ARM Cortex-A9 based CA9 NEC chip, clocked at 400 MHz. The CA9 NEC chip has limited power management functions without power gating or DVFS, which needs to be noted when considering the power consumption of the system. As powering on and off cores is the main power reduction technique available on this particular chip, the power consumption can not be precisely matched to the required performance. In a system where power gating and DVFS is available, the possibilities for energy proportional computing are better. A Debian installation with a compatible Linux kernel version 2.6.28 is provided with the Versatile Express. Official support for the Versatile Express in the Linux kernel was not added before version 2.6.33. The operating system was installed on a USB flash drive, as the native memory card on the Versatile Express is significantly slower than the USB flash drive.

The Versatile Express allows monitoring of both operating voltage and power consumption for the evaluated processor. To use this functionality, a kernel module was created, and loaded into the kernel running on the Versatile Express, in order to access the necessary register for collecting voltage, current, and power consumption data. The register used for this is VD10 _ S3, the power measurement device for the Cortex-A9 system supply, cores, MPEs, SCU, and PL310 logic. A program that reads the values for voltage, current, and power once every second and store them will be used. The use of the program allows continuous monitoring during benchmarking.

### B. Tegra

A Tegra 200 series developer kit [12] is used to evaluate the performance of the Tegra 250 chip. The Tegra 250 chip includes a dual core Cortex-A9 MPCore chip running at 1 GHz. The board also contains 1 GB of DDR2-667 RAM. The Tegra board has an additional PCI express Gigabit Ethernet card installed in order to avoid networking bottlenecks. By evaluating the performance of both the Tegra and the CoreTile Express, the aim is to identify how the number of cores, and the clock frequencies are reflected in the performance for running different applications. Linux kernel version 2.6.32, which was provided by Nvidia, is used on the Tegra. As the only compatible kernel version available for the Tegra is 2.6.32, and the Versatile Express is not supported before 2.6.33, the two Cortex-A9 systems cannot use the same kernel version. For tests that required a larger space for file storage that could not be stored on the memory card, a external USB hard drive was used. While the performance of a external USB hard drive is modest, so is the performance of the test system in comparison to the servers that we are comparing to.

No information on the power consumption of the Tegra 250 chip is available, thereby the values used are estimates, derived from the information released by ARM [13]. The Tegra 250 chip does not provide information about energy consumption and as the chip includes several specialized processors, such as a graphics processor, it is not possible to physically measure the energy consumption of the Cortex-A9, on the development kit. According to ARM a Dual Core Cortex-A9 built using the TSMC (Taiwan Semiconductor Manufacturing Company) 40G process uses 1.9 W at 2 GHz in a performance optimized implementation, and 0.5 W at 800 MHz in a power optimized implementation. In this paper, a estimate of 1 W is used for the Cortex-A9 in the Tegra 250 chip.

## VI. Benchmarks

Benchmarks will be used to evaluate the performance of our hardware. First, we will use Autobench to evaluate the performance of the Apache 2 HTTP server by measuring the number of small static web pages the different hardware is capable of serving. In order to gather information about how different hardware performs with slightly more demanding web services, SPECweb2005 will be used. The Erlang run time system (rts) will be benchmarked using a set of micro benchmarks to see how well the rts performs on the hardware. This will enables us to evaluate how well an application running on top of Erlang could be expected to run. To model a realistic service scenario, a Erlang based SIP-Proxy will also be tested.

### A. *Apache*

The Apache 2.2 HTTP server was used to determine how well the Cortex-A9 MPCore machines can perform with more traditional server tasks. The Apache server was chosen as it is freely available, open source, and has been one of the most popular servers for a long time. The Apache HTTP server is available for many platforms such as Linux and Mac OS/X, and is used with a variety of hardware architectures. As these benchmarks are targeting the x86 and ARMv7-A architectures, the ability for the Apache HTTP server to run on both is crucial. The first tests are focusing on use cases with small static files. Later we will move on testing more demanding web services.

Autobench [14] is used to measure how well Apache performs on the different machines, and with different test parameters. Autobench is a tool that helps automate the use of httperf. Httperf in turn is a program for benchmarking the performance of an HTTP server. It creates connections to a server in order to fetch a file. During one connection, one or several requests for the file can be made depending on the test parameters. By changing the rate at which the connections are created, and the number of requests for each connection, the load on the server varies. By running httperf several times with different rates of connections it can be evaluated how the server responds to different loads.

In the autobench tests when benchmarking a machine with two Intel Quad Core Xeon E5430 processors, the test proved to not be CPU intensive enough as full CPU utilization was not achieved. In order to find the bottleneck the number of clients was increased. The test was done using both ten and five client machines. As the results were the same in both tests, the performance of the client machines was not a bottleneck. The bandwidth was tested by redoing the test using a larger file than the original. The result from the test with the larger file was close to that of the original test, the biggest exception being a higher bandwidth usage. The system reported no shortage of available memory in any of the Apache HTTP server benchmarks. As neither bandwidth, CPU, or the client machins were bottlenecks, the remaining possibility is that the performance at this point is memory bound. This is an example of the previously mentioned problem with performance un balance between different components in servers. A problem that could be avoided by using less powerfull processors.

| Machine | Request / s | Requests / J |
|---|---|---|
| Quad Core Intel Xeon E5430 (2.66 GHz) | 33000 | 413 |
| Pentium 4 (2.8GHz) | 7100 | 80 |
| Dual Core Cortex-A9 MPCore (1 GHz) | 4600 | 4600 |
| Quad Core Cortex-A9 MPCore (400 MHz) | 3400 | 2833 |
| Cortex-A8 (600 MHz) | 760 | 760 |

Table I
ABILITY OF APACHE 2.2 TO SERVE A 10 BYTE STATIC FILES USING DIFFERENT HARDWARE

The machine with the two Quad Core Xeons was able to serve 36000 requests per second when one hundred requests were made for each connection. For ten requests per connection the result was only 6200 requests/s. The server running the Apache server reported 60 % CPU utilization for the test with 36000 requests, and 10 % for the test with 6200 requests. If the CPU utilization level and web server performance would continue having the same relation to each other, the performance in both cases is around 60000 requests per second with full CPU utilization. Assuming the performance for one E5430 running at 100 % is the same as the performance of two running at 50 %, the performance for one E5430 is 33000 requests per second.

The results from the Autobench shown in Table I are from fetching a static file of size 10 Bytes. 10 Calls per connection and 100 calls per connection were requested in the tests. The better results from the two benchmarks were used. The performance for the Tegra 250 was more or less the same when making ten or a hundred requests for each connection. For the comparison machine with the two Xeon processors the difference was approximately a multiple of ten. As can be seen in Table I the Tegra 250 managed to serve 4600 requests per second, and the Versatile Express 3400 requests per second. The performance difference of the Versatile Express compared to the Tegra 250 is likely caused by both the slower clock frequency of the CPU and the network implementation. The difference in combined clock frequencies between the two processors on its own is slightly less than the performance difference. The combined number of clock ticks for the Versatile Express is 1600 (400 * 4) and for the Tegra 250 2000 (1000 * 2). Comparing these, the Versatile Express has 80 % of the clock tick of the Tegra 250. The performance of the Versatile Express is in comparison slightly less, 74 % of that of the Tegra 250.

To provide a more comprehensive comparison and more reference points, a machine with a Pentium 4 (2.8 GHz) was also benchmarked. While the machine that has the more traditional server processors outperforms the tested Cortex-A9 processors, the Cortex-A9 processors do well taking their energy consumption into account. The Intel Xeon processor (E5430) that was used in the reference machine has a reported TDP of 80 W, while the Quad Core Cortex-A9 according to performed tests has a maximum measured power consumption of 1.2 W.

The rightmost column in Table I shows the number of answered request served per Joule for the Autobench test. A clear improvement in energy efficiency is visible starting from the Pentium 4 to the Dual Core ARM Cortex-A9 MPCore. Figure 2 shows the energy efficiency comparison as a bar diagram, and indicates a energy efficiency of about 6,9 times the performance per Joule for the Versatile Express compared to the Intel Xeon. The energy efficiency of the Tegra 250 compared to the energy efficiency of the reference Intel Xeon processor, was approximately 11,1 times greater. A clear improvement in energy efficiency is also visible between the Pentium 4 processor and the Xeon. This improvement is an indication on the energy efficiency improvement for Intel's x86 based processors. One of the major improvements from the Pentium 4 to the Xeon L5430 is the manufacturing technology that has improved from 90 nm to 45 nm.

To evaluate the performance of the Tegra 250 for more demanding, and more realistic web services, the SPECweb2005 benchmark was used. SPECweb2005 consists of three different workloads; ecommerce, banking and support [15]. The ecommerce workload emulates a web based shopping system, the banking workload a online banking system, and the support workload a web page that emulates a hypothetical support web page. The results from the individual benchmarks are the number of simultaneous user sessions that the systems can support. During the benchmarks both SSL and regular, simple connections are created. The actions of the simulated users are varied between the different user sessions during the test. In addition the amount of user sessions, the quality of service is measured. Predefined quality of service requirements must be met for all benchmarks. The definitions for what is considered a good or tolerable results, are defined separately for the three benchmarks. In all cases, 95 % of the requests made must meet the requirements for 'good' quality of service, and 99 % for 'tolerable'. In order for testrun to be valid, certain minimum running times must be achieved, every test is run several times and the results compared. Small differences are allowed but in cases where the results vary too much between the runs, the results are considered invalid. Using the results from the three different tests, and a set of reference results the final SPECweb score is calculated. The reason for using the SPECWeb2005 benchmark is that it is the most standardised benchmark with published results for comparison, that we were able to run on our test systems. Most of the comparison results provided on the SPECweb2005 results page are from machines running Rock Web Server or Zeus Web Server [16]. As these servers do not support ARM based processors they can not be used in these measurements. There are a few results from machines running Apache and PHP that we can compare our results to. The same versions of Apache and PHP were installed on our test machines as was used on the reference results. The Apache web server version 2.2.9 and PHP version 5.2.6 were used. One of the reference machines used PHP 5.1.6, but on the Tegra no performance difference was found between these two versions, so the PHP 5.2.6 was used for the Tegra.

| Machine | Ecommerce | Banking | Support |
|---|---|---|---|
| Quad Core Intel Xeon X3360 (1) | 3600 | 2700 | 4200 |
| Quad Core Intel Xeon X3360 (2) | 7360 | 6240 | 7840 |
| Dual Core Cortex-A9 MPCore (1 GHz) | 230 | 180 | 220 |

Table II
NUMBER OF SIMULTANEOUS SESSIONS USING DIFFERENT HARDWARE

| Machine | Ecommerce | Banking | Support |
|---|---|---|---|
| Quad Core Intel Xeon X3360 (1) | 38 | 28 | 44 |
| Quad Core Intel Xeon X3360 (2) | 77 | 66 | 83 |
| Dual Core Cortex-A9 MPCore | 230 | 180 | 220 |

Table III
NUMBER OF SIMULTANEOUS SESSIONS SUPPORTED FOR EACH USED WATT

For comparisons on how many simultaneous user sessions in the three different use cases the Tegra 200 series development kit could serve in comparison with more powerfull server machines, a machine with a Quad Core Intel Xeon X3360 processor was used [16]. The TDP of a X3360 is 95 W [17]. To decrease bottlenecks caused by the machines disk drives there were separate disks for server logs, and operating system, as well as 120 disks in raid 0 for the files that were to be served [16]. In addition another server with the same processor but only 24 disks for the web pages and PHP 5.1.6 was used. As can be seen from Table II the X3360 can sustain upp to 33 times the number of sessions compared to the Tegra 250. Comparing these results to those from the autobench test the differences in performance is much greater. Reasons for this is found in the added complexity of the tasks and the added disk drive performance of the reference machine. It is higly likely that the test results we are comparing against in this test have been well optimized, as they are created by the servers manufacturer. Compared to the reference machine with less disk drives the performance difference is much smaller. The performance of the Tegra in this test is likely to have doubled just through PHP caching and the usage of a faster disk drive, instead of a memory card.

The energy efficiency for the Banking, Ecommerce and Support tests can be seen in Table III. Although the difference in energy efficiency for the Xeon based processor and the ARM based processor is not as great as it was in the Autobench test, it is still remarkable. Combining the results from all three tests the ARM processor is 2.9 times

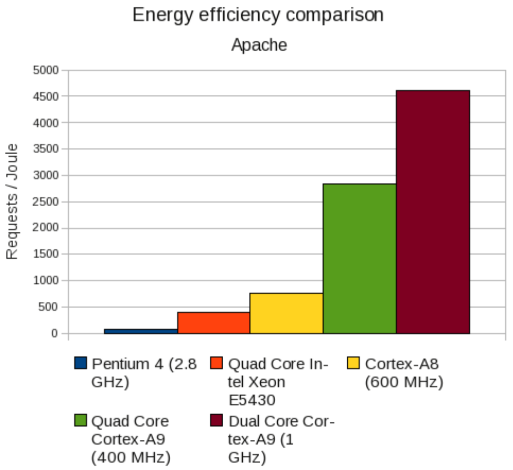

Figure 2. Number of requests handled for each Joule used by the CPU

more energy efficient. One important detail is that in the last test the ARM based test server had no optimization at all. By using the results from these Apache HTTP benchmarks together with Hamiltom's cost model, we can conclude that a data center could save up to 12,7 % of its total lifetime costs by using these more energy efficient processors.

### B. Erlang based SIP-Proxy

Erlang [18], [19] is a functional programming language designed to be highly concurrent and suited for fault tolerant soft real-time systems. Erlang run time system (rts) implements its own lightweight processes and garbage collection mechanism. The execution of Erlang processes is controlled by one or more schedulers. One scheduler is generally used for each available core, and all schedulers are run as separate processes on the host operating system. There is no shared memory between Erlang processes, which means that all communication is done using message passing, facilitating the construction of distributed systems.

SIP [20] is an application-layer protocol for controlling sessions with one or more participants. It is used for creating, modifying, and terminating sessions. The sessions can be multimedia, including video or voice calls, and the session modification possibilities include the ability to add or remove media and participants, or change addresses. The protocol itself can be run on top of several different transport protocols including the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). SIP enables the creation of an infrastructure consisting of proxy servers that users can use to access a service. A SIP-proxy is a server that helps route requests to the current location of the user and makes requests on behalf of the client.

An Erlang based SIP-Proxy was tested in order to evaluate how well an ARM Cortex-A9 would perform in telecom applications. The performance of the SIP-Proxy was measured in the number of calls per second it could handle. The metric for energy efficiency of the proxy was selected to be the number of calls the proxy could handle for each Joule used. The measurement setup requires two additional PCs. One PC will be used as the sender and the other will be used as the receiver. Both of these machines are running SIPp, an open source test and traffic generation tool made available by HP [21]. The version used is 3.1 and is compiled from the source code.

The x86 reference machine used for comparison purpose has two Quad-Core Intel Xeon L5430 processors running at 2.66GHz. The test result for the reference machine is presented in Figure 3. In cases where the CPU is the bottleneck, the performance increase is approximately dependent on the amount of CPU resources available, in this case, the number of cores. As can be seen from Figure 3 this is not the case here. There is a significant performance improvement all the way from one scheduler (SMP1) to four schedulers (SMP4). When the number of schedulers is increased from four to eight there is only an increase of 50 calls/s, although the number of available schedulers has doubled enabling the usage of all eight cores. As the focus in this test is on processor performance, and energy efficiency, results that are not dependent on the processors themselves are discarded. With the added burden of the performance unbalance the reference server would performs much worse. Only the results from one to four cores will be used, in practice treating the reference machine as having only one Quad-Core processor, using only the energy required by one rather than two processors. An issue with comparing the energy efficiency in this test is that the only energy consumption data available is the Thermal Design Power (TDP) information provided by the manufacturer. According to the information available on the Intel web page the TDP of the L5430 is 50 W, and that it is manufactured using a 45 nm process [10]. When evaluating the performance of the SIP-Proxy the Versatile express was able to handle 30 calls/s. The reference machine with its Intel Xeon L5430 was able to handle 350 calls/s. Both machines were tested using increasing numbers of schedulers (1-4). These results can be seen in Table IV. By taking into account that the CPU of the reference machine has a maximum TDP of 50 W, compared to the measured maximum consumption of 1.2 W used by the Cortex-A9, the Cortex-A9 performs well. By comparing the throughputs and the power consumptions, it can be seen that the Cortex-A9 can handle 3.6 times more traffic for each watt it dissipates, compared to the Intel Xeon. In this comparison the energy consumption listed for the Xeon is according to the manufacturer rated TDP. As there is no actual energy consumption data available for the Xeon, the maximum measured energy consumption for the Quad Core Cortex-A9 is also used rather than the actual consumption.

In other benchmarks the Tegra 250 has consistently over performed the CoreTile Express except in this, and a TCP

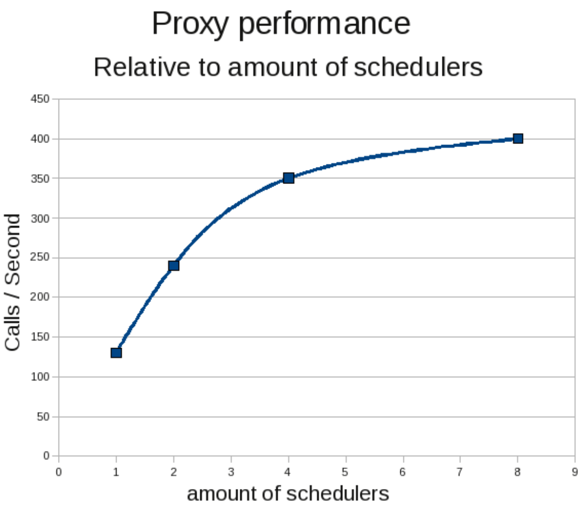

Figure 3. Performance of reference machine with two Quad Core Xeons with increasing number of schedulers

| SMP | Intel Xeon<br>L5430 50 W | Quad Core Cortex-A9<br>(400 MHz) 1.2 W | Dual Core Cortex-A9<br>(1 GHz) 1 W |
|---|---|---|---|
| 1 | 130 | 5 | 5 |
| 2 | 240 | 12 | 13 |
| 4 | 350 | 30 | 13 |

Table IV
How many calls the SIP-Proxy can handle using the Intel Xeon and the Cortex-A9 processors

message benchmark, performed as part of the preliminary Erlang benchmarking. While the Versatile has the advantage of having double the number of CPU cores the Tegra 250 has more than double the clock frequency on its cores. As visible in Table IV, the performance of the Versatile Express and the Tegra 250 is very similar when using the same number of cores. If we were only using one core the difference could be due to a static overhead of running the proxy, but the results from when using two cores are not as easily dismissible. The main question here is why an almost identical performance increase is achieved, when adding a core running at 400 MHz and one at 1000 MHz.

As the performance of the proxy when running on the Tegra 250 was not as expected from the technical data available, and our previous benchmarks, additional steps to certify the results were taken. The Erlang rts on both the Tegra 250 and the CoreTile Express was recompiled from the same source, and in the same way, using the same version of GCC, and the same version of the libatomic library (7.2 alpha 4). As this did not cause any difference in the results the Erlang rts was again recompiled with a few changes to support profiling using Gprof. The performance on the CoreTile Express was affected more by running Gprof than on the Tegra 250. A test was then performed using five calls per second and SMP 2 for two minutes on both machines, while profiling using Gprof. The biggest difference between the number of function calls and time spent in different functions was in functions that have to do with atomic read functions. This is caused by the fact that the schedulers are frequently left without work and at that point, in an attempt to optimize spins over a variable to check for more work. To match the throughput between the two test machines, the Tegra was not under maximum load causing the schedulers to be without work more often than on the V2P-CA9. Other significant differences were not observed. In order to profile system wide rather than just the Erlang rts, the kernels on both the Tegra 250 and the CoreTile Express were recompiled to support Oprofile [22]. Oprofile showed that when running on as high load as possible, the Tegra 250 spent 31,6 % of its time running vmlinux, while the CoreTile Express spent 20,6 %. The times spent running the Erlang rts were 65 % and 67 % respectively. The remaining possible explanations for the performance anomaly is a a bug in the kernel used for the Tegra, that in combination with the proxy or (Erlang rts) caps the performance, or as the tested platform is rather new, and generally not used in this manner, some unexpected hardware issue could also be causing this. However, the tests that were run on the Erlang rts, were meant to find issues like this one. As the performance difference data for the proxy was not backed up by the Erlang rts tests, hardware problems are not a likely cause.

To find out the cost savings for a data center we will again use Hamilton's model. Using the results from the SIP-Proxy results we find a cost saving potential of 10 % by the usage of more energy efficient processers, such as the Cortex-A9.

## VII. Conclusions

The performance of two test machines with ARMv7 based ARM Cortex-A9 processors, running a Apache 2.2 HTTP server, and an Erlang based SIP-Proxy, has been measured and compared to the performance of a few different servers with Intel Xeon processors. The focus was on the energy efficiency of the processors. The energy efficiency for the Dual Core Cortex-A9 compared to the E5430, while running the HTTP serve,r was up to 11.1 times greater, enabling a total cost saving for a data center of 12.7 %. In addition the energy efficiency of the Quad Core Cortex-A9 compared to a L5430 was 3.6 times greater when running a Erlang based SIP-Proxy, enabling a 10.0 % decrease in the total cost for a data center.

The x86 processors that have been benchmarked are not the newest available server processors. Newer processors are in general more energy efficient than older processors, decreasing the difference between the new ARM based processors, and the most energy efficient x86 based processors. However even with doubled energy efficiency from the x86 processors, the difference to the ARMv7 processors is remarkable.

Further research is needed to determine how several of the less powerful processors should be connected as a simple SMP architecture is only suitable for a limited amount of processors. One option would be a cloud on a chip solution. The development of such as system requires research to determine a variety of design decisions such as communication paths and hierarchies, as well as the optimal amount of processors in the chip.

## REFERENCES

[1] J. Hamilton, "Cooperative expendable micro-slice servers (cems): Low cost, low power servers for internet-scale services," in *Proceedings of CIDR 09*, January 2009.

[2] "Overall data center costs. http://perspectives.mvdirona.com/2010/09/18/overalldatacentercosts.aspx," James Hamilton, September 2010, online; accessed 14 March 2011.

[3] L. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, December 2007.

[4] D. A. Patterson, "Latency lags bandwidth," *Communications of the ACM*, vol. 47, October 2004.

[5] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," 2007, iSCA '07 Proceedings of the 34th annual international symposium on Computer architecture.

[6] C. Belady, A. Rawson, J. Pfleuger, and T. Cader, "The green grid power efficiency metrics: Pue % dcie," 2008, online; accessed 31 January 2011. [Online]. Available: http://www.thegreengrid.org/ /media/WhitePapers/White _Paper6 _- _PUE _and _DCiE _Eff _Metrics _30 _December _2008.ashx?lang=en

[7] R. Miller, "Microsoft embraces data center containers. http://www.datacenterknowledge.com/archives/2008/04/01/microsoft-embraces-data-center-containers," April 2008, online; accessed 16 June 2011.

[8] S. Shankland, "Google uncloaks once-secret server. http://news.cnet.com/8301-1001 _3-10209580-92.html," April 2009, online; accessed 16 June 2011.

[9] R. Miller, "Microsoft: 300,000 servers in container farm. http://www.datacenterknowledge.com/archives/2008/05/07/microsoft-300000-servers-in-container-farm/," May 2008, online; accessed 16 June 2011.

[10] *Quad-Core Intel Xeon Processor 5400 Series Datasheet*, Intel, August 2008, online; accessed 18 January 2011. [Online]. Available: http://www.intel.com/Assets/en _US/PDF/datasheet/318589.pdf

[11] *CoreTile Express A9x4 Cortex-A9 MPCore (V2P-CA9) Technical Reference Manual.*

[12] Nvidia, *Nvidia Tegra 200 series developer kit, quick start guide*, December 2009, dU-04942-001v02.

[13] ARM, "Cortex-a9 processor. http://www.arm.com/products/processors/cortex-a/cortex-a9.php," online; accessed 3 January 2011.

[14] "Autobench web site. http://www.xenoclast.org/autobench/," online; accessed 16 January 2011.

[15] (2011, February) Specweb2005. Standard Performance Evaluation Corporation. [Online]. Available: http://www.spec.org/web2005/

[16] (2010, September) Specweb2005. Standard Performance Evaluation Corporation. [Online]. Available: http://www.spec.org/web2005/results/web2005.html

[17] *Intel Xeon Processor 3300 Series datasheet volume 1*, Intel Incorporated, February 2009. [Online]. Available: http://www.intel.com/Assets/PDF/datasheet/319005.pdf

[18] "Erlang programming language official web site. http://www.erlang.org/faq/introduction.html," online; accessed 16 June 2011. [Online]. Available: http://www.erlang.org/faq/introduction.html

[19] J. Armstrong, *Programming Erlang, Software for a Concurrent World*, 2007th ed. Raleigh, North Carolina Dallas, Texas: Pragmatic Bookshelf, 2007.

[20] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "Sip: Session initiation protocol," June 2002, online; accessed 1 February 2011. [Online]. Available: http://tools.ietf.org/html/rfc3261

[21] "Sipp web site. http://sipp.sourceforge.net/," online; accessed 17 March 2011.

[22] "Oprofile web page. http://oprofile.sourceforge.net/," online; accessed 15 March 2011.